

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-020495

(43)Date of publication of application : 29.01.1993

(51)Int.Cl.

G06K 9/20

G06F 15/62

G06F 15/66

G06F 15/66

G06K 9/00

(21)Application number : 03-171133

(71)Applicant : CANON INC

(22)Date of filing : 11.07.1991

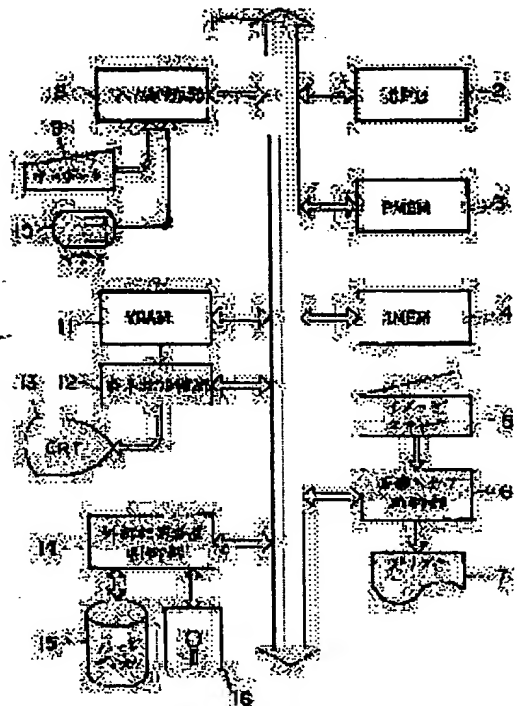
(72)Inventor : HARA HIROYUKI

## (54) IMAGE PROCESSOR

### (57)Abstract

**PURPOSE:** To convert a subject image all at once into the data of different forms for each part of the image when the image data of a certain form is converted into the data of another form.

**CONSTITUTION:** The image data stored in an IMEM 4 are displayed on a CRT 13. An operator designates a part that is converted into a character code after recognition of a character and a part that is converted into the graphic data based on the displayed image data. When the start of conversion is instructed, those designated parts are converted into each designated form and stored in an idle area of a PMEM 5 or the IMEM 4. When the conversion is completed, the converted data are read out from the storage destination and synthesized with the recomposed image data. Then the synthetic data are displayed on the CRT 13.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(51) Int. Cl.<sup>8</sup> F I

G 0 6 K 9/20 戸内整理番号  
G 0 6 F 15/62 3 4 0 L 9073-5 L  
15/66 3 2 0 A 8125-5 L  
3 3 0 Q 8420-5 L  
4 0 0 8420-5 L  
G 0 6 K 9/00 K 9289-5 L

審査請求 未請求 請求項の枚数 3

(52) 出願番号 特願平 3-171133

(52) 出願日 平成 3 年 (1991) 7 月 11 日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子 3 丁目 30 番 2 号

(72) 発明者 原 寛行

東京都大田区下丸子 3 丁目 30 番 2 号 キヤノ

ン株式会社内

(74) 代理人 弁理士 大塚 康樹 (外 1 名)

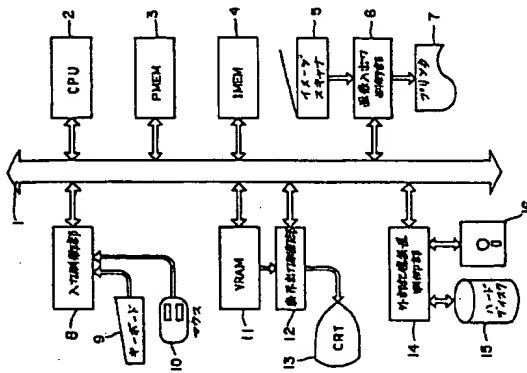
(金 11 頁)

(54) 【発明の名称】 画像処理装置

(57) 【要約】

【目的】 1 つの画像データを他の形式のデータに変換する際、変換対象の画像を、その部分ごとに異なる形式のデータに一度に変換できるようにする。

【構成】 I M E M 4 に格納されている画像データは C R T 1 3 に表示されている。オペレータは表示されている画像データから、文字認識して文字コードに変換する部分と、図形データに変換する部分とを指定する。変換開始が指示されると、先に指定された領域を指定された形式に変換し、P M E M 3 または I M E M 4 の空き領域に格納しておく。変換が終了したら、変換後のデータを格納先から読み出し、イメージデータを再構成してそれを合成し、C R T 1 3 に表示する。



【特許請求の範囲】

【請求項 1】 画像データを他の形式に変換して格納しておく画像処理装置であって、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 1 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 2 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 3 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 4 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 5 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 6 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 7 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 8 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 9 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 10 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 11 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 12 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 13 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 14 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 15 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 16 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 17 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 18 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 19 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 20 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 21 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 22 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 23 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 24 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 25 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 26 の指定手段と、

前記装置が、前記画像データを他の形式に変換する際のその形式を指定する第 27 の指定手段と、

(2)

の入力装置がスキャナであった場合には、読み込み度、画像がわずかにあるが変化してしまうため、正確な変換がしづらいという問題点があった。

【0004】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0005】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0006】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0007】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0008】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0009】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0010】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0011】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0012】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0013】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0014】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0015】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0016】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0017】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0018】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0019】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0020】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0021】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0022】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0023】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0024】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0025】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0026】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0027】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0028】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0029】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

【0030】 また上述した装置においては、画像データを一度も読み直さずという問題点があった。

- (3) する条件を具体的に示した図である。図中、B・C・D・Eは分岐点であり、この点で連続する点列は別れる。すなわち本図の場合、連続する点列(以下、線分と呼ぶ)はA・B・C・D・E・F・B・G・C・H・D・I・Eの9個であり、これらの線分が各々独立した図形データに変換される。例えば分岐点Bに注目すると、線分B・Gの長さが図4で設定した条件に合致したと判定して、線分A・BまたはB・Cの長さが図4で設定した条件に合致していれば、それに沿って図形に変換される。
- 10 [0010] 図8は表示画像データ21内で文字データに変換したい領域を指定した図であり、図9の説明と同様の指定方法でマウスカーソル31を用いて指定する。このようにして指定した領域が01・02・03・04であり、図形変換領域34と区別するために線分で示される。図9では文字変換領域は図形変換領域の内側に存在しているが、これは説明の都合上内側にきたためで、実際には画像データ内のどこにあつてもかまわない。
- 20 [0017] 図7は画像データ内の図形変換領域34と文字変換領域01・02・03・04を各々図形データおよび文字データに変換したあとの状態を示した図である。図中71・72・73・74・75は文字データに変換された領域で、それ以外は図形データに変換されている。また、図8の画像データで表示されていた図7に示された、図8の画像データに変換された領域に前記の図形変換条件により図形データに変換された図形変換領域であり、このような画像データは白く消去されている。
- 30 [0018] 次に、変換例の画像処理システムの動作について、フローチャートを用いて説明する。
- [0019] 図8および図9・図10は本実施例の動作を説明するフローチャートであり、CPU2によって実行される処理の手順を示したものである。図8のフローチャートは図形変換・文字変換を行なうまでの画像データに対する変換領域指定の動作を説明しており、図9のフローチャートは図形変換の処理過程を、図10のフローチャートは文字変換を参照におこない、その結果をCRT13上に表示するまでの動作を説明している。これらは便宜上分割されているが一連の処理である。また図示したのは変換処理にかかわる部分だけで、関係ない部分は省いてある。
- 40 [0020] まずオペレータは、図形をイメージスキャナ5から読み込ませる。
- [0021] ステップS1において、イメージスキャナ5から読み込んだ画像の画像データをIMEM4に格納し、VRAM11に展開してCRT13に画像データを表示する。
- 50 [0022] 次に、オペレータは図形変換領域を指定するために図形変換条件指定をメニューよりマウス10を用いて選択する。メニューが選択されたら、選択された内容についてテストする。
- [0023] まずステップS2において、選択されたメニューが図形への変換であるか文字への変換であるかをテストする。図形への変換であれば、ステップS3で条件となる領域を指定させる。即ち、オペレータは画像データ内の任意位置でマウスの左ボタンの押下を開始し、押下の状態のままマウスを移動すると、上記押下開始位置と現在のマウスカーソル位置を対角線とすると矩形が画線で描かれる。この矩形内の領域が図形データ変換領域となり、この領域はマウスの左ボタンの押下を終了することにより確定される。図形変換領域は直線でかまわない。
- [0024] ステップS4は、上記図形変換領域を指定した時に表示される条件設定ウインドウ(図4に示す)に対し、図形データへの変換条件を入力させるステップである。オペレータはウインドウの指示に従って条件を入力し、確定を指定することにより図形変換条件が確定する。この変換条件指定ウインドウは図形変換領域ひとつについて1回表示されるため、各変換領域に対し異なる条件を設定することができる。
- [0025] ひとつの条件設定が終わると、再びメニューから項目を選択する。その選択内容をテストし(ステップS7・S2)、文字への変換条件の設定であれば、ステップS5で変換領域の指定を行なう。すなわち、オペレータが画像データ内の任意位置でマウスの左ボタンの押下を開始し押下の状態のままマウスを移動すると、押下開始位置と現在のマウスカーソル位置を対角線とする矩形が画線で描かれる。線分で描かれるのは、図形変換領域と区別できるようにしたためである。この矩形内の領域は文字データ変換領域となり、この領域はマウスの左ボタンの押下を終了することにより確定される。文字変換領域は直線でかまわない。
- [0026] ステップS6は、文字変換領域が指定された時に表示される文字変換条件決定ウインドウに対し、指定した領域内の文字が横書きか、縦書きか、スペースを認識するか等の文字データの識別条件を入力させるステップである。ウインドウの指示にしたがってオペレータは認識条件を入力し、確定を指定することにより、文字認識条件が確定される。この文字認識条件決定ウインドウは変換領域ひとつについて1回表示されるため、各変換領域に対し異なる文字認識条件を設定することができる。
- [0027] 図形変換領域および文字変換領域の指定が完了した後、またメニューから項目を選択させる。その選択内容をテストし(ステップS7)、変換処理開始が選択されている。変換の指定された変換領域に対し変換処理を開始する。変換の内訳的にまずおこなう処理は、文字変換領域があるかどうかの判断であり、いいかえれば操作者がステップS5・0を行なったかどうかであり、図により求めた変換点をスプライン曲線の制御点と
- (4) する。このステップがステップS8である。
- [0028] ステップS8において、文字変換領域が有いと判断した場合には、ステップS9・ステップS11は行わず、図形変換処理にそのまま進む。
- [0029] ステップS7において、文字変換領域が有と判断した場合には、次のステップS9に進み、P)EM3もしくはIMEM4内の空いた領域に文字変換のバッファを新たに作り、そこに文字変換領域内の画像データをコピーする。このコピーバッファは文字変換データひとつに対しひとつ存在している。
- 10 [0030] コピー終了後、ステップS10において、画像データ内の文字変換領域を白く消去する。これは、変換の際に、文字変換領域内の画像データを図形に変換するのを防ぐ意味と、図形変換のスピードを速くする意味をもっている。
- [0031] 以上、条件の設定が終わったなら、ステップS11から実際の図形変換・文字変換処理に入る。
- [0032] 図9のフローチャートについて説明する。
- [0033] まずステップS11で、変換対象の図形指定された変換領域があるかをテストする。もしあるならば、ステップS12において図形変換領域内の画像データの細分化処理を行なう。細分化処理とは、画像データを1ドット幅の線画データにする処理である。本実施例の画像処理システムでは、図形の認識の前段階の処理として細分化のアルゴリズムを用いているが、この処理手法は一手法であって、これ以外の手法を採用していても問題はない。
- 20 [0034] 図形変換領域内の画像データの細分化処理が終わした後、ステップS13において、図形変換領域内を走査し、画像データの連続した点列を1組抽出する。
- [0035] 次に、ステップS14において、ステップS13で抽出された連続した点列が、ステップS4で定めた図形変換条件に合うかどうかを判断する。この条件を満たしていると判断した場合には、次のステップS16に進み、満たしていなければステップS15に入り、次の連続した点列の抽出を行なう。
- [0036] ステップS14で条件を満たしていれば、ステップS15において、ステップS13で抽出された点列について、角となる点(以下、角点と呼ぶ)を抽出する。この角点の抽出は、本実施例では角の曲率を求めることによりおこなっている。ただし、それ以外の手法であつても問題ない。
- 30 [0037] ステップS16においては、角点を含ん分けられる組の点列の一方が直線でもう一方が曲線であるという様に、点列の形状が異なつていて認識し、結合する。その角点上で上記連続する点列を分割する。
- [0038] さらに、ステップS17において、曲線認識された点列に対しては、まず折れ線近似し、その図により求めた変換点をスプライン曲線の制御点と

から得られた文字の画像を重ね合わせて新たな画像データを生成し、それを表示することである。

【0049】以上の手順で、もとの画像から得られた図形データと文字データを得、得られたデータから画像を再現することができ、また、変換時に条件を指定することで、文字画像が変換された図形データに変換されることなく、図形データの情報量が文字画像をも図形データに変換した場合に比べ少なくて済むという効果がある。また、図形データへの変換と文字データへの変換を同時に行うことにより、変換処理が効率化するという効果もある。

【0050】

【他の実施例】

【実施例2】変換例1においては、文字画像データの図形データへの変換を防ぐことを主な目的として、指定した長さ以上の連続した点列のみを図形データに変換するという簡単な条件付けをさせている。これに文字のポインタ数を条件として加えることにより、指定ポインタ数から得られる1文字の領域内に、連続する長さが指定されている、連続する点列が操作者の指定した長さ以上のデータであつても、図形データへの変換は行なわないということができる。こうすることにより、文字データと図形データの切り分けが一層明確化される。

【0051】また、上記説明におけるポインタ数の指定にかえて、縦横の長さ（ドット数でもmmでも可）を直接指定することでも可能である。例えば、表示画像データ内の適当な1文字を囲むような領域を、マウス10により指定することで自動的に1文字分の縦横の長さを求めるようにしても可である。マウスにより直線を描くことによつて（ただし、この直線は図形データとは関係ない）、描かれた直線から求める長さを条件としてもよい。

【0052】また、実施例1においては、操作者が指定した長さ以上の連続する点列を図形データに変換するようにしているが、「以上」と「以下」を操作者が指定できるようにしてもよい。これによつて、「以下」を指定することにより細かい画像データのみを図形データに変換させることが可能となる。これは上記で述べたポインタ数の指定においても同様である。

【0053】実施例1においては、一通の操作はマウス毎のポインタインデックスを用いることにより行なっているが、キーボードから操作できるようにしてもよい。

【0054】また、実施例1においては、画像データの入力はイメージスキャナからおこなっているが、画像データが取り込められれば、他の入力装置からでも、外部記憶装置に記憶された画像データであつても構わない。

【0055】また、図形データおよび図形データに変換する画像データの領域指定は、矩形領域で行なっている

が、円・楕円・多角形などの閉領域であつてもよい。

【0056】実施例1においては、変換後の図形データの表示する際、図形データ自身の情報量が小さく、変換前の画像データを消去しているが、画像データを消去せず残しておいてもよいし、画像データ表示消去の切り替え手段を設けることにより、操作者が任意のタイミングで表示・消去を行なうようにしてもよい。

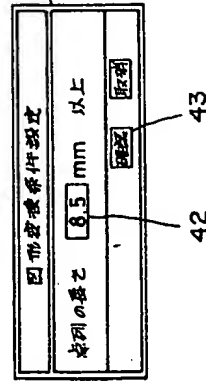
【0057】【実施例3】実施例1・2の装置では、1つの画像を文字の部分と図形の部分に分け、それぞれ文字・図形のデータ形式に変換するものであった。しかし変換はこの2種に限ったものでなく、他の変換が盛じられてもよい。

【0058】実施例1の文字・図形データへの変換に加えて、無変換という変換を行う。即ち、画像データそのまゝの領域を指定させ、図11において、1点領域で囲まれた領域が画像データそのまゝであると指定された部分である。このように第3の変換をするならば、まず変換条件の指定のメニューにもう1種追加する必要がある。即ち、図8のメニュー3～5と並んで無変換領域の指定をさせる処理が追加される。また、ステップ10で文字変換領域をとりだした後、無変換領域の有無をテストし、有るならば、その領域をとりだしてメモリの空き部分に退避し、画像データから無変換領域111を消去してしまう。無変換であるから変換にかかわる処理は特にない。画像を再現する際に、図形データから再生された画像と文字データから再生された画像の合成が行なわれてから、無変換領域として保存されていた領域111を更に合成してビットマップデータを作り、CRT13に表示する。

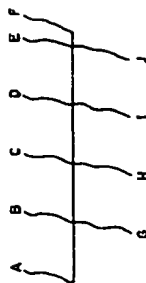
【0059】この様に、3種類の処理を取り混ぜて行ない、変換されたデータから画像を再構成する装置でも実施例1と同様に処理することができ、

【0060】尚、本発明は複数の機器から構成されるシステムに適用しても、1つの機器からなる装置に適用してもよい。また、本発明はシステム或は装置にプログラムを供給することによって達成される場合にも適用できることは言うまでもない。

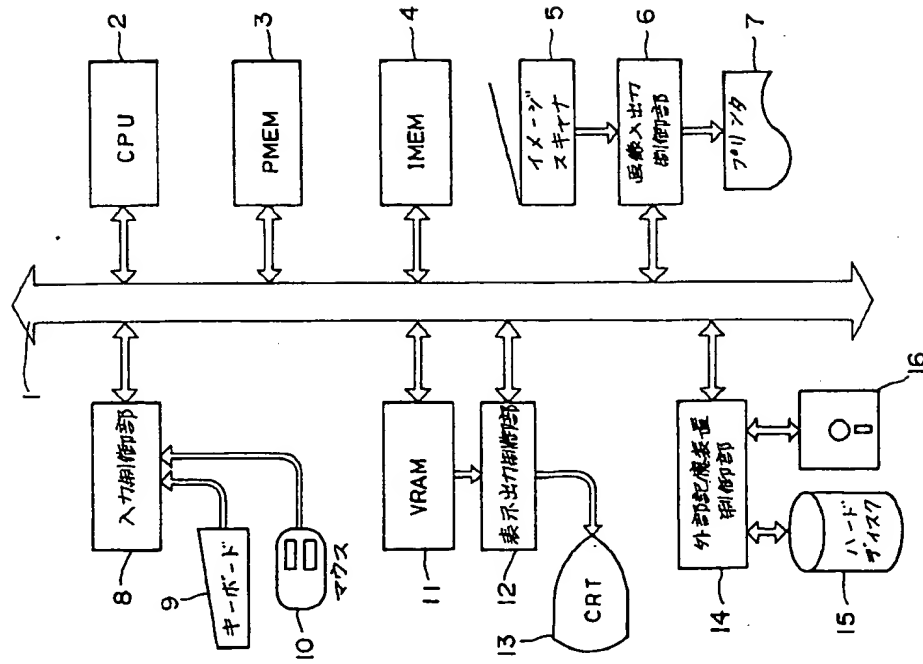
【図4】



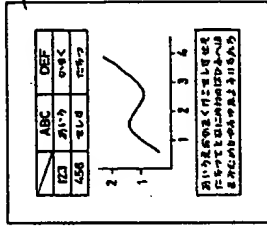
【図5】



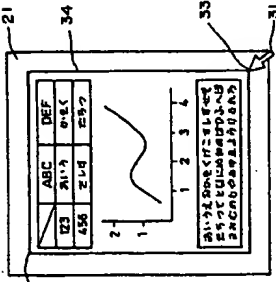
【図1】



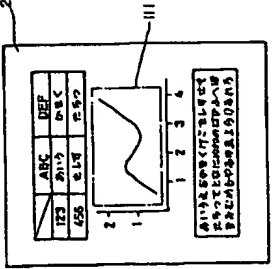
【図2】



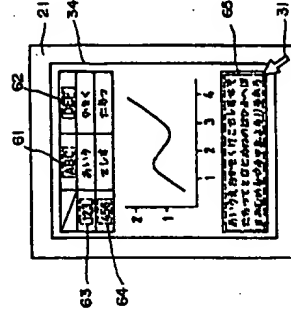
【図3】



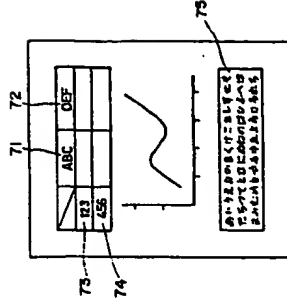
【図11】



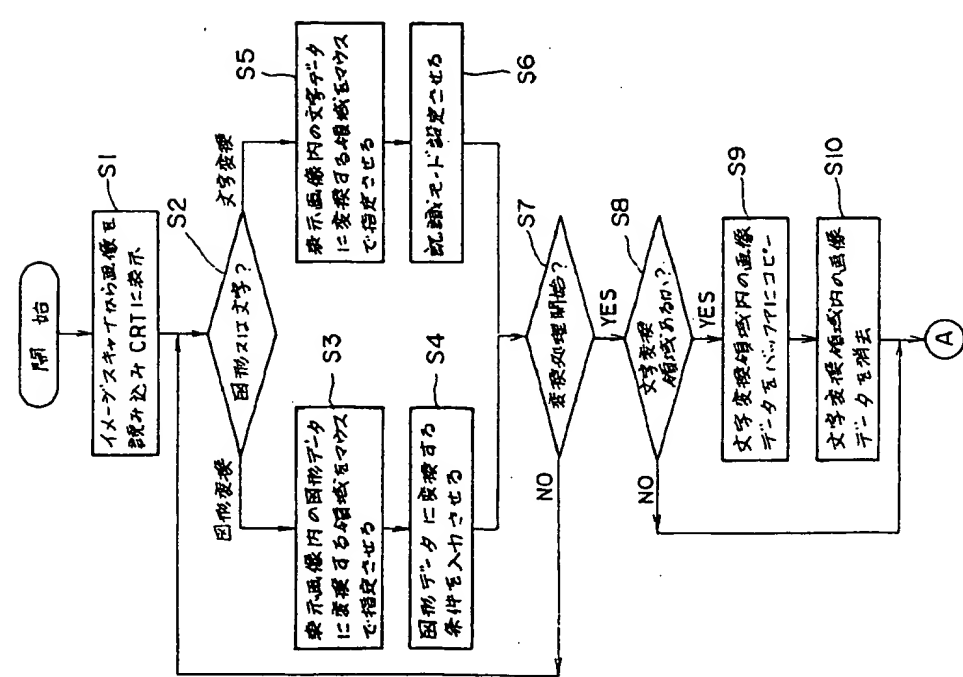
【図6】



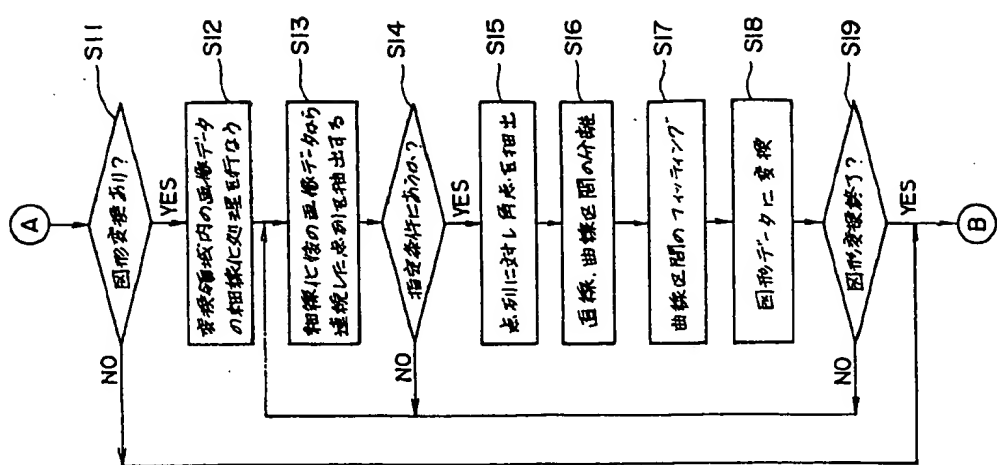
【図7】



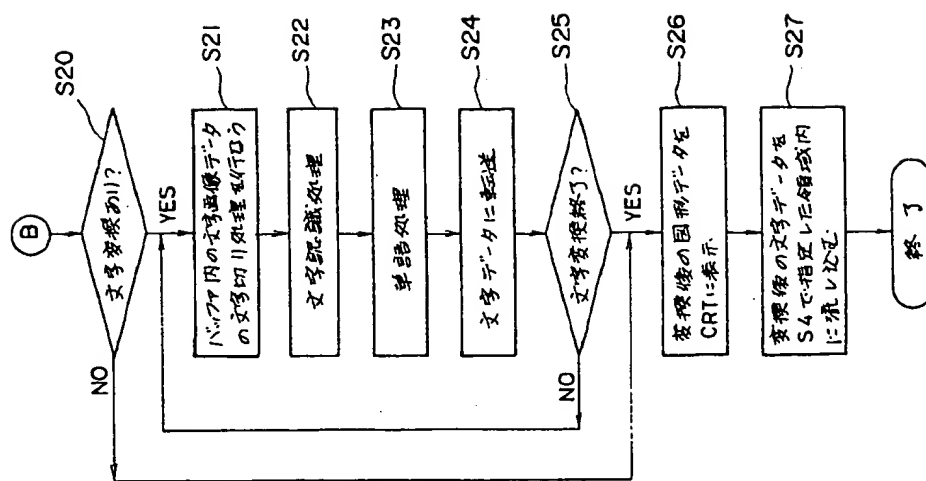
【図8】



【図9】



【図10】



\* NOTICES \*

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

---

DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Industrial Application] This invention relates to the image processing system which reads and carries out data conversion of the image.

[0002]

[Description of the Prior Art] If it was in the equipment which changes one image data into the data of a different class like graphic data and alphabetic data conventionally, each conversion function existed independently like the function to change image data into graphic data, and the function to change image data into alphabetic data. For this reason, in telling graphic data that a portion with one image changes another portion into alphabetic data, in order to obtain graphic data first, an image is read from a scanner etc., and after editing the read image and removing an unnecessary portion, it changes into graphic data, and memorizes the obtained graphic data temporarily. Next, in order to obtain alphabetic data, reading repair and the read image are again edited from a scanner etc., and said image data is changed into alphabetic data. Thus, the original image has been reproduced only after performing each conversion independently, reproducing image data from the obtained data and re-compounding them.

[0003]

[Problem(s) to be Solved by the Invention] However, in the equipment mentioned above, in order to read image data in order to change a certain portion in one image data into alphabetic data to change a certain another portion into graphic data at alphabetic data, and to change into graphic data, the same image data must be reread again. Thus, the trouble of [ whenever the class of conversion changes, it is necessary to reread an image, although images are few whenever the trouble that processing becomes complicated, and the input unit for reading image data read into a \*\*\*\*\* case with a scanner, in order to change, exact conversion carries out, and ] \*\*\*\*\* is \*\*\*\*\*.

[0004] Moreover, the trouble that image edit etc. must be performed to the image data whenever it reads since only a required field is transformed to graphic data or alphabetic data, and there is nothing in the equipment mentioned above in addition to the time and effort of rereading image data repeatedly if it is \*\*\*\* is \*\*\*\*\*.

[0005] This invention was made in view of the above-mentioned conventional example, can perform at once different conversion for every portion specified to one image data, and aims at offering the image processing system which can reconfigure the original image easily from the data of a different class changed and acquired.

[0006]

[Means for Solving the Problem] In order to attain the above-mentioned purpose, an image processing system of this invention consists of the following configurations.

[0007] The 1st assignment means which specifies the format at the time of being the image processing system which changes and stores image data in other format, and changing image data into other data format, The 2nd assignment means which specifies a field changed into format specified with this assignment means in image data as 1 set of formal assignment means It has a conversion means to



change into data format which had image data in a field specified by two or more sets of formal assignment means which are two or more format that data format specified with said 1st assignment means differs, and said formal assignment means specified.

[0008]

[Function] The image processing system of this invention divides one image data into some portions by the above-mentioned configuration, and it changes into the data of different format for every portion of the. Moreover, the changed data can be returned to image data and they can be compounded.

[0009]

[Example] the following and an accompanying drawing -- therefore, the example of this invention is explained to details. In addition, if the function of this invention is performed, it cannot be overemphasized in the system to which processing is performed also for \*\*\*\*\* through networks, such as LAN, by the system which \*\*\*\*\* also becomes from two or more devices by the device of a simple substance that this invention is applied also for \*\*\*\*\*.

[0010] [Example 1] drawing 1 is system block drawing of the image processing system which is the example of this invention. 1 is a system bus and each configuration block to be explained from now on is connected to this system bus. 2 is CPU (Centrtal Processing Uint). 3 is memory (Following PMEM is called) which stores a program or a text. Selection/reading of the program for edit processing are suitably done from a hard disk 14 at PMEM3, and it is performed by CPU2. Moreover, the data inputted from the keyboard 9 is stored in PMEM3 as code information. The graphic data changed from image data are also stored in PMEM. After [ VRAM / 11 ] being developed by IMEM of 4, it is developed upwards, and the image data into which 4 is a printer and image memory (Following IMEM is called) and 5 were read [ 4 / an image I/O control unit and 7 ] for an image scanner and 6 from the image scanner 5 is displayed on CRT13. Moreover, a printing image is outputted to a printer 7 by control of image I/O control unit 6, after being developed by IMEM4 as bit map data. 8 is the input-control section and the input unit of keyboard 9 and mouse 10 grade is connected. An operator performs the operating command of a system etc. by operating this keyboard 9. Moreover, the mouse 10 is for carrying out selection / processing directions of the image information on CRT13, and in this example, although it is using the mouse, it should just be a pointing device. The mouse cursor on CRT13 is moved in the direction of X-Y with a mouse 10 at arbitration, and selection and edit of selection and the graphic data of the selection and the image data of a menu are performed. As for video image memory (VRAM is called hereafter) and 12, 11 is [ a display-output control section and 13 ] CRT. The data displayed on CRT13 is developed by VRAM11 as bit map data. For example, if it is graphic data, the graphic form pattern corresponding to its location and attribute information is developed by VRAM as an image. Moreover, it is possible to generate direct cursor to the data of VRAM by software control, and to display. 14 is an external storage control section and a disk for data files in 15-16, 15 is a hard disk (Following HD is called), and 16 is a floppy disk (Following FD is called).

[0011] Drawing 2 to drawing 7 is drawing explaining the activity of the image processing system of this example.

[0012] It is drawing having shown the condition of drawing 2 having read the image from the image scanner 5, and having displayed on CRT13, and 21 is display image data.

[0013] Drawing 3 is drawing which specified the field to change into graphic data within the display image data 21. The field 34 made by moving pushing the left carbon button of a mouse to the termination location 33 of a field to change the cursor (for a mouse cursor to be called below) 31 which shows the location on CRT13 of a mouse 10 from the starting position 32 of a field to change turns into a conversion field.

[0014] Drawing 4 is drawing having shown the screen for setting up the conditions at the time of changing the image data in the above-mentioned conversion field 34 into graphic data. 41 is a graphic form conversion conditioning window, and only when a numeric character is inputted into the condition input area 42 in a window 41 from a keyboard 9 and the length of the sequence of points which image data follows is said beyond input-numeric character, it is changed into graphic data. When inputting nothing, all the image data in a graphic form conversion field is changed into graphic data. The

conditions of graphic form conversion will be determined by specifying a confirmation button 43.

[0015] Drawing 5 is drawing having shown concretely the conditions which change continuous sequence of points into graphic data. B-C-D-E is the branch point among drawing, and the sequence of points which continue at this point go out. That is, in this Fig., continuous sequence of points (it is hereafter called a segment) are nine of AB-BC-CD-DE-EF-BG-CH-DI-EJ, and these segments are changed into the graphic data which became independent respectively. for example, when the branch point B is observed, it is \*\*\*\*\* to the conditions which the length of Segment BG set up by drawing 4 -- and if the conditions which the length of Segments AB or BC set up by drawing 4 also as \*\* are suited, it will be changed into a graphic form noting that the intermediary \*\*\*\* segment BG fulfills [ the rope ] conditions to it.

[0016] Drawing 6 is drawing which specified the field to change into alphabetic data within the display image data 21, and is specified using a mouse cursor 31 with the same specification method as explanation by drawing 3. Thus, the specified field is 61-62-63-64, and it is shown by the dashed line in order to distinguish from the graphic form conversion field 34. Although the transliteration field exists inside a graphic form conversion field in drawing 6, this is because it came inside on account of explanation, and \*\*\*\*\* does not matter in fact where in image data, either.

[0017] Drawing 7 is drawing having shown the condition after changing respectively the graphic form conversion field 34 and the transliteration field 61-62-63-64 in image data into graphic data and alphabetic data. The inside 71-72-73-74-75 of drawing is the field changed into alphabetic data, and is changed into graphic data except it. Moreover, in case the image which is not shown in drawing 7 while being displayed by the image data of drawing 6 is changed into graphic data, it is changed into graphic data according to the aforementioned graphic form conversion conditions, and it is inside \*\*\*\* image data and such image data is eliminated white.

[0018] Next, actuation of the image processing system of an example is explained using a flow chart.

[0019] Drawing 8, and drawing 9 and drawing 10 are the flow charts explaining actuation of this example, and show the procedure of the processing performed by CPU2. The flow chart of drawing 8 explains actuation of the conversion block definition to image data until it performs graphic form conversion and a transliteration, the flow chart of drawing 9 performs the processing process of graphic form conversion, the flow chart of drawing 10 actually performs a transliteration, and actuation until it displays the result on CRT13 is explained. Although these are divided for convenience, they are a series of processings. Moreover, what was illustrated is only a portion in connection with transform processing, and the portion which is not related is excluded.

[0020] An operator makes a graphic form read from an image scanner 5 first.

[0021] In step S1, the image data of the image read from the image scanner 5 is stored in IMEM4, it develops to VRAM11, and image data is displayed on CRT13.

[0022] Next, in order to specify a graphic form conversion field, an operator uses a mouse 10 and chooses graphic form conversion criteria specification from a menu. If a menu is chosen, it will test about selected contents.

[0023] It first tests whether in step S2, the selected menu is conversion in a graphic form, or it is conversion to an alphabetic character. If it is conversion in a graphic form, the target field will be made to specify at step S3. That is, if an operator starts the depression of the left carbon button of a mouse in the arbitration location in image data, and moves a mouse with the condition of a depression, and the above-mentioned depression starting position and a current mouse cursor location are made into the diagonal line, a rectangle will be drawn as a continuous line. The field in this rectangle turns into a graphic-data conversion field, and this field is decided by ending the depression of the left carbon button of a mouse. \*\*\*\*\* does not care about a graphic form conversion field with plurality, either.

[0024] Step S4 is a step into which the conversion conditions to graphic data are made to input to the conditioning window (shown in drawing 4) displayed when the above-mentioned graphic form -- conversion field is specified. an operator -- directions of a window -- therefore, conditions are inputted and graphic form conversion conditions are decided by specifying a check. this conversion conditioning window -- even [ a graphic form conversion field ] -- \*\*\*\*\* -- since the first half of the 1st inning is

shown, \*\*\*\*\* conditions can be set up to each conversion field.

[0025] After one conditioning finishes, an item is made to choose from a menu again. The contents of selection are tested (step S7, S2), and if it is a setup of the conversion conditions to an alphabetic character, an object domain will be made to specify at step S5. That is, if an operator starts the depression of the left carbon button of a mouse in the arbitration location in image data and moves a mouse with the condition of a depression, the rectangle which makes the diagonal line a depression starting position and a current mouse cursor location will be drawn with a dashed line. Because it enabled it to distinguish from a graphic form conversion field, it is drawn with a dashed line. The field in this rectangle turns into an alphabetic character data-conversion field, and this field is decided by ending the depression of the left carbon button of a mouse. \*\*\*\*\* does not care about a transliteration field with plurality, either.

[0026] Step S6 is a step into which the discernment conditions of whether the alphabetic character in the field specified to the character recognition condition decision window displayed when a transliteration field is specified recognizes lateral writing, columnar writing, and a space, and alphabetic data are made to input. Character recognition conditions are decided by an intermediary operator's inputting recognition conditions, although it was made directions of a window, and specifying a check. this character recognition conditioning window -- even [ a conversion field ] -- \*\*\*\*\* -- since the first half of the 1st inning is shown, different character recognition conditions to each conversion field can be set up.

[0027] After assignment of a graphic form conversion field and a transliteration field is completed, an item is made to choose from a menu. If the contents of selection are tested (step S7) and transform-processing initiation is chosen, transform processing will be started to the specified conversion field. The processing internally performed first in the case of conversion is decision whether there is any transliteration field, and, in other words, is whether the operator performed step S5 and 6. This step is step S8.

[0028] In step S8, when it is judged that there is no transliteration field, step S9 and step S10 are not performed, but progresses to graphic form transform processing as it is.

[0029] In step S7, when it is judged that there is a transliteration field, it progresses to the following step S9, the buffer for transliterations is newly taken to the field which \*\*ed in PMEM3 or IMEM4, and the image data in a transliteration field is copied there. Even a transliteration field receives and this one copy buffer exists.

[0030] In step S10, the transliteration field in image data is eliminated white after copy termination. the semantics which prevents this incorrect-recognizing the image data in a transliteration field in a graphic form in the case of graphic form conversion, and the semantics which makes speed of graphic form conversion quick -- with, it is.

[0031] As mentioned above, if a setup of conditions is completed, actual graphic form conversion and transliteration processing will be started from step S11.

[0032] The flow chart of drawing 9 is explained.

[0033] It tests whether at step S11, there is any conversion field specified as the graphic form for conversion first. Supposing it is, in step S12, thinning of the image data in a graphic form conversion field will be performed. Thinning is processing which uses image data as the line drawing data of 1-dot width of face. Although the algorithm of thinning is used as processing of the preceding paragraph story of recognition of a graphic form in the image processing system of this example, this thinning technique is a way method, and it is satisfactory even if it has adopted technique other than this.

[0034] After thinning of the image data in a graphic form conversion field is completed, in step S13, the inside of a graphic form conversion field is scanned, and 1 set of sequence of points which image data followed are extracted.

[0035] Next, in step S14, the continuous sequence of points extracted at step S13 judge whether the graphic form conversion conditions set up by step S4 are suited. When it is judged that conditions are fulfilled here, if it is not progressing and filling to the following step S15, the sequence of points by which return and a degree followed step S13 are extracted.

[0036] If conditions are filled with step S14, in step S15, the point (the salient point is called hereafter) used as an angle will be extracted about the continuous sequence of points extracted at step S13. The extract of this salient point is \*\*\*\* intermediary \*\*\*\* by asking for the curvature of sequence of points in this example. However, \*\*\*\*\* is also satisfactory by any technique other than this.

[0037] In step S16, in a straight line, when one side of 2 sets of sequence of points divided on both sides of the salient point has recognized it as intermediary \*\*\*\* \*\*\*\* in the configuration of sequence of points so that it may say that another side is a curve, it divides the above-mentioned sequence of points which carry out continuation in the salient point.

[0038] Furthermore, in step S17, to the sequence of points recognized to be a curve, polygonal-line approximation is carried out first and a curve is approximated for the folding point which was able to be found by the approximation as a control point of a spline curve. Of course, as for the curve to approximate, a problem does not have \*\*\*\*\* in any way not at a spline curve but at a Bezier curve etc., either.

[0039] At step S18, conversion to actual graphic data is performed based on the contents of recognition to step S17. It is changed into three sorts of graphic data of a straight line, the polygonal line, and a curve in this example. Let the approximation function with which the polygonal line passes along both ends and each top-most vertices, and a curve passes along representation points (control point) and those points two points to which a straight line is located in the both ends be graphic data. The obtained graphic data are stored in the free area of PMEM3 or IMEM4. Things other than the three above-mentioned kind may be added as a class of graphic data changed.

[0040] In step S19, when it judges whether all the data that carried out thinning was changed into graphic data and all are not changed yet, it repeats from extract processing of the sequence of points of step S13.

[0041] Conversion to graphic data is completed above. Next, explanation about transliteration processing of drawing 10 is performed.

[0042] It tests whether the field which should be carried out a transliteration is specified as a step of the beginning of character recognition (step S20).

[0043] If it is, alphabetic character end processing will be performed to the image data in the buffer copied by step S9 in step S21.

[0044] Next, in step S22, recognition processing of an alphabetic character is performed about the image by which the alphabetic character end was carried out.

[0045] In step S23, word collating with the result in step S22 which carried out character recognition, and a word dictionary is performed.

[0046] In step S24, it changes into alphabetic character (code) data based on the character recognition result in step S22. The changed data of a result is stored in the free area of PMEM3 or IMEM4.

[0047] In step S25, it judges whether transform processing to all transliteration images was completed. When it is judged that it has not ended yet, the processing from step S21 is repeated until conversion to a character code ends about all the candidates for conversion. When it is judged that graphic form transform processing was completed at step S19, and transliteration processing was completed at step S25, it progresses to step S26. At this step S26, from the graphic data after conversion, an image is reconfigured, the bit map data for a display is created, and it stores in VRAM11. This image is displayed on CRT13.

[0048] In step S27, alphabetic data [ finishing / a transliteration ] is slushed into a void \*\*\*\*\* field in the field S10 which specified the transliteration at step S5, i.e., a step, inside, and the result is displayed on CRT13. This is laying the image of the alphabetic character obtained from character code data on top of the bit map data of VRAM11, creating new image data to it, and displaying it on it.

[0049] In the above procedure, the graphic data and alphabetic data which were obtained from the image of a basis can be obtained, and an image can be reproduced from the obtained data. Moreover, it is lost by specifying conditions at the time of conversion that an alphabetic character image is easily changed into graphic data, and it is effective in it being few compared with the case where the amount of information of graphic data also changes an alphabetic character image into graphic data, and ending.

Moreover, there is also an effect that transform processing increases the efficiency, by performing conversion to graphic data, and conversion to alphabetic data to coincidence.

[0050]

[Other Example(s)]

In the [example 2] example 1, easy conditioning of changing only the sequence of points followed more than the specified length by setting it as the main purposes to prevent conversion to the graphic data of alphabetic character image data into graphic data is carried out. If the sequence of points which continue in the field of one character obtained from an assignment point size by adding to this a condition [ the point size of an alphabetic character ] are settled, it will become possible for the conversion to graphic data not to perform \*\*\*\*\* by the data more than the length which the operator specified [ continuous sequence of points ], either. By carrying out like this, alphabetic data and graphic data carving is clarified further.

[0051] Moreover, it is also possible to change to assignment of the point size in the above-mentioned explanation, and to specify length (for the number of dots or mm to be sufficient) in every direction directly. For example, it is good for making it find the length in every direction for one character automatically by specifying a field which surrounds one suitable character in display image data with a mouse 10, and drawing a straight line with a mouse also considering the length which can be found from the drawn straight line therefore (however, this straight line is not related to graphic data) as conditions.

[0052] Moreover, although he is trying to change into graphic data the sequence of points followed more than the length specified by an operator, an operator may enable it to specify the "above" and the "following" in an example 1. It becomes possible by therefore specifying the "following" as this to transform only fine image data to graphic data. This is the same also in assignment of the point size described above.

[0053] Although a series of actuation is performed by using pointing devices, such as a mouse, you may enable it to operate it from a keyboard in an example 1.

[0054] Moreover, in an example 1, \*\*\*\*\* does not care about an image entry of data with the image data it was remembered to be by external storage even from other input units if only intermediary \*\*\*\* from an image scanner could incorporate image data, either.

[0055] Moreover, although the block definition of the image data changed into graphic data and graphic data is performed in the rectangle field, \*\*\*\*\* is also good in closed regions, such as a circle, an ellipse, and a polygon.

[0056] Although the own check of graphic data has eliminated the image data before conversion so that easily in case the graphic data after conversion display, it may leave image data as it is, and an operator may enable it to perform display and elimination to the timing of arbitration by establishing the change means of image data display elimination in an example 1.

[0057] It was what divides one image into the portion of an alphabetic character, and the portion of a graphic form, and changes it into the data format of an alphabetic character and a graphic form with the [example 3] example 1 and the equipment of 2, respectively. However, conversion is not what was restricted to these two sorts, and other conversion may be mixed.

[0058] In addition to conversion to the alphabetic character and graphic data of an example 1, conversion of no changing is performed. namely, image data -- a field as it is is made to specify the field surrounded with the dashed line in drawing 11 -- image data -- it is the portion specified that it remains as it is. Thus, if 3rd conversion is carried out, it is necessary to add one more sort to the menu of a setup of conversion conditions first. That is, the processing which makes a non-changed field specify together with steps S3-S6 of drawing 8 is added. Moreover, if the existence of a non-changed field is tested after taking out a transliteration field at step 10, the field will be taken out, it will evacuate to the empty portion of memory, and the non-changed field 111 will be eliminated from image data. Since it did not change, there is especially no processing in connection with conversion. In case an image is reproduced, after composition of the image reproduced from graphic data and the image reproduced from alphabetic data finishes, the field 111 saved as a non-changed field is compounded further, bit map data is made, and it displays on CRT13.

[0059] Thus, it can process with the equipment which mixes three kinds of conversion together, carries out, and reconfigures an image from the changed data as well as an example 1.

[0060] In addition, even if it applies this invention to the system which consists of two or more devices, it may be applied to the equipment which consists of one device. Moreover, it cannot be overemphasized that it can apply also when therefore attained by that this invention supplies a program to a system or equipment.

[0061]

[Effect of the Invention] As explained above, the image processing system concerning this invention can perform at once different conversion for every portion specified to one image data, and can reconfigure the original image easily from the data of a different class changed and acquired.

---

[Translation done.]